



日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 7月13日

出 願 番 号

Application Number:

特願2000-212482

出 願 人

Applicant (s):

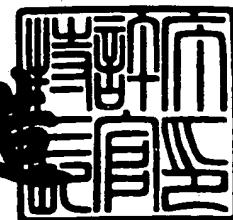
富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年12月22日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3105649

【書類名】 特許願

【整理番号】 0051429

【提出日】 平成12年 7月13日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/00

【発明の名称】 拡大鍵生成装置および記録媒体

【請求項の数】 6

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 下山 武司

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 伊藤 孝一

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 武仲 正彦

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 鳥居 直哉

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 矢嶋 純

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

株式会社内

【氏名】 屋並 仁史

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通

株式会社内

【氏名】 横山 和弘

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100089141

【住所又は居所】 東京都目黒区平町 1 丁目 2 1 番 2 0 - 6 0 3 号

【弁理士】

【氏名又は名称】 岡田 守弘

【電話番号】 03-3725-2215

【手数料の表示】

【予納台帳番号】 015543

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705795

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 拡大鍵生成装置および記録媒体

【特許請求の範囲】

【請求項 1】

暗号鍵から拡大鍵を生成する拡大鍵生成装置において、

入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数 i 回それぞれ行なって複数 i の演算結果を生成し、これら生成した各グループ毎の複数 i の演算結果について複数のグループ間で該当演算結果をそれぞれ 1 つにまとめる演算を行ない、複数 i の中間データを生成する中間データ生成手段と、

指定された拡大鍵の段数 r をもとに、上記複数 i の中間データから 1 つを選択し、選択した中間データを非可逆変換して段数 r の拡大鍵を生成する拡大鍵生成手段と

を備えたことを特徴とする拡大鍵生成装置。

【請求項 2】

上記入力された暗号鍵のビット列として、入力された暗号鍵のビット列に非線型関数を演算したビット列としたことを特徴とする請求項 1 記載の拡大鍵生成装置。

【請求項 3】

上記 1 つにまとめる演算として、論理演算としたことを特徴とする請求項 1 記載の拡大鍵生成装置。

【請求項 4】

上記 1 つにまとめる演算を行なった後、非線型関数を演算して中間データを生成したことを特徴とする請求項 1 記載の拡大鍵生成装置。

【請求項 5】

上記選択した中間データについて段数 r に従った転置を行なった後に非可逆変換を行なうことを特徴とする請求項 1 記載の拡大鍵生成装置。

【請求項 6】

入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グ

ループのビット列に演算を複数 i 回それぞれ行なって複数 i の演算結果を生成し、これら生成した各グループ毎の複数 i の演算結果について複数のグループ間で該当演算結果をそれぞれ 1 つにまとめる演算を行ない、複数 i の中間データを生成する中間データ生成手段と、

指定された拡大鍵の段数 r をもとに、上記複数 i の中間データから 1 つを選択し、選択した中間データを非可逆変換して段数 r の拡大鍵を生成する拡大鍵生成手段と

して機能させるプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、暗号鍵から拡大鍵を生成する拡大鍵生成装置および記録媒体に関するものである。

【0002】

【従来の技術】

一般的な共通鍵暗号による暗号化処理の構成を図 8 に示す。図 8 の暗号化装置は、平文と暗号鍵を入力とし、暗号文を出力するものであって、暗号化処理装置と拡大鍵生成装置とから構成されている。暗号化処理装置は、暗号化処理 1 から暗号化処理 n の順番に、 n 段階の処理を行なう。拡大鍵処理装置は、入力された暗号鍵をもとに、 n 段階の暗号化処理のそれぞれで用いられる拡大鍵 1 から拡大鍵 n の生成を順次行なう。この拡大鍵の生成は、重要な問題であり、高速性と安全性とが要求されている。

【0003】

従来、高速処理可能な方式として、DES の拡大鍵生成方式がある。この DES の拡大鍵生成方式は、図 9 の右側の拡大鍵生成装置に示すように、暗号鍵を入力とし、巡回シフトとビット転置のみにより拡大鍵 1 から拡大鍵 n の生成を行なうため、処理が高速である。

【0004】

また、安全性がより高い方式として、MARS の拡大鍵生成方式 (MARS-a

candidate cipher for AES, THE First AES Conference 1998 p1-p9)がある。

【0005】

【発明が解決しようとする課題】

上述した前者の図9のDESの拡大鍵生成方式は、循環シフトとビット転置のみ（図9の右側の*の部分参照）により拡大鍵の生成を行なうため、処理が高速である反面、拡大鍵のビット情報から暗号鍵のビット情報を容易に得ることができてしまうという問題がある。このため、n個の拡大鍵のうち1個でも情報が漏洩した場合、暗号鍵の情報まで漏洩することとなり、安全性に問題があった。

【0006】

また、上述した後者のMARSの拡大鍵生成方式は、拡大鍵の情報から暗号鍵の情報を簡単に得ることができないので、安全性が高い反面、多くの演算を繰り返すため、高速に処理を行なうことができないという問題があった。

【0007】

本発明は、これらの問題を解決するため、第1段階で暗号鍵から中間データを生成し、第2段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成し、任意段の拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることを目的としている。

【0008】

【課題を解決するための手段】

図1を参照して課題を解決するための手段を説明する。

図1において、暗号化装置1は、平文および暗号鍵を入力とし、暗号文を出力するものであって、拡大鍵処理装置3などから構成されるものである。

【0009】

拡大鍵処理装置3は、暗号鍵から拡大鍵を生成するものであって、ここでは、中間データ生成装置4および拡大鍵生成装置5などから構成されるものである。

中間データ生成装置4は、暗号鍵を入力とし、複数iの中間データを生成するものである。

【0010】

拡大鍵生成装置 5 は、複数 i の中間データから指定された段数 r の拡大鍵を生成するものである。

次に、動作を説明する。

【0 0 1 1】

中間データ生成装置 4 が入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数回 i それぞれ行なって複数 i の演算結果を生成し、これら生成した各グループ毎の複数 i の演算結果について複数のグループ間で該当演算結果をそれぞれ 1 つにまとめる演算を行ない、複数 i の中間データを生成し、拡大鍵生成装置 5 が指定された拡大鍵の段数 r をもとに複数 i の中間データから 1 つを選択し、選択した中間データを非可逆変換して段数 r の拡大鍵を生成するようにしている。

【0 0 1 2】

この際、入力された暗号鍵のビット列として、入力された暗号鍵のビット列に非線型関数を演算したビット列とするようにしている。

また、1 つにまとめる演算として、論理演算を行なうようにしている。

【0 0 1 3】

また、1 つにまとめる演算を行なった後、非線型関数を演算して中間データを生成するようにしている。

また、選択した中間データについて段数 r に従った転置を行なった後に非可逆変換を行なうようにしている。

【0 0 1 4】

従って、第 1 段階で暗号鍵から中間データを生成し、第 2 段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成することにより、拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることが可能となる。

【0 0 1 5】

【発明の実施の形態】

次に、図 1 から図 7 を用いて本発明の実施の形態および動作を順次詳細に説明する。

【0016】

図1は、本発明のシステム構成図を示す。

図1において、暗号化装置1は、平文および暗号鍵を入力とし、暗号文を出力するものであって、暗号化処理装置2および拡大鍵処理装置3などから構成されるものである。

【0017】

暗号化処理装置2は、拡大鍵1から拡大鍵nをもとに暗号化処理(1)から暗号化処理(n)のn段階の処理を行ない、暗号文を作成して出力するものである。暗号化処理(1)から暗号化処理(n)は、拡大鍵処理装置3で生成された拡大鍵1から拡大鍵nを受け取ってそれぞれの暗号化処理を行ない、最終段から暗号文を出力する。

【0018】

拡大鍵処理装置3は、暗号鍵から拡大鍵を生成するものであって、ここでは、中間データ生成装置4および拡大鍵生成装置5などから構成されるものである。

中間データ生成装置4は、暗号鍵を入力とし、複数iの中間データを生成するものである(図2、図3など参照)。

【0019】

拡大鍵生成装置5は、複数iの中間データから指定された段数rの拡大鍵を生成するものである(図2、図4など参照)。

次に、図2のフローチャートの順番に従い図1の構成のもとで暗号鍵から拡大鍵を生成するときの動作を詳細に説明する。

【0020】

図2は、本発明の動作説明フローチャートを示す。

図2において、S1は、ユーザ鍵を入力する。これは、図1の暗号化装置1に平文と一緒に、暗号鍵(ユーザ鍵)を入力する。これにより、図1の拡大鍵処理装置3を構成する中間データ生成装置4がユーザ鍵(暗号鍵)を取り込んだこととなる。

【0021】

S2は、非線型関数Mを演算する。これは、後述する図3に示すように、暗号

鍵のビット列を8グループに分割し、各グループのビット列に非線型関数Mを演算する（図6、図7を用いて後述する）。

【0022】

S3は、偶数のときに定数を加算する。

S4は、奇数のときに定数を乗算する。これらS3、S4は、後述する図3に示すように、非線型関数Mを演算した後のビット列に、偶数のときは定数を加算し、奇数のときは定数を乗算する。

【0023】

S5は、排他的論理和を演算する。これは、後述する図3に示すように、S4で定数を加算したビット列と、定数を乗算したビット列とを排他的論理和の演算を行なう。

【0024】

S6は、非線型関数Mを演算する。これは、S5で演算した後のビット列に、非線型関数Mを演算し、中間データを生成する。尚、S4、S5、S6で、複数の定数（例えば図3の場合には $i = 0, 1, 2$ に対応する3つ定数）についてそれぞれ加算、乗算した後に、排他的論理和演算、更に非線型関数演算をそれぞれ行ない、図3の場合にはそれぞれ3つからなる中間データを生成する。

【0025】

以上のS1からS6によって、ユーザ鍵（暗号鍵）を入力とし、例えば図3の構成に従い、非可逆変換を経た複数つつ（図3では3つつ）からなる中間データを作成することが可能となる。

【0026】

図2のS7は、段数 r を入力する。これは、作成しようとする、拡大鍵の段数 r を入力する。これにより、本発明では、指定した段数の拡大鍵を以降のS8からS11で直接に作成することが可能となる。

【0027】

S8は、中間データから該当する値を選択する。これは、後述する図4の（a）の複数 i つつの中間データ中から、段数 r をもとに該当する中間データを1つつそれぞれ選択する。

【0028】

S 9 は、段数 r に従った転置を行なう。これは、図 4 の (a) のデータ並び替え処理装置で、段数 r を入力して、S 8 で選択した中間データの転置を行なう（図 5 の (c), (d) 参照）。

【0029】

S 1 0 は、転置後の中間データの非可逆変換 G を行なう（図 4 の (b) 参照）。

S 1 1 は、 r 段数の拡大鍵を出力する。これは、S 1 0 で演算した後のデータを r 段数の拡大鍵として出力する。

【0030】

S 1 2 は、終わりか判別する。YES の場合には、終了する。NO の場合には、S 7 に戻り、次の指定された段数 r の拡大鍵の作成を行なう。

以上の S 7 から S 1 1 によって、S 1 から S 6 で作成した複数 i づつの中間データをもとに、指定された段数 r に従って 1 つづつの中間データを選択し、段数 r に従って転置した後、非可逆変換を行い、指定された段数 r の拡大鍵を非可逆変換を経て高速に生成することが可能となる。以下順次詳細に説明する。

【0031】

図 3 は、本発明の説明図（中間データ）を示す。

図 3 において、上段の $k 0$ から $k 7$ は、暗号鍵（ユーザ鍵）のビット列を順次 8 分割したビット列である。

【0032】

上段および下段の M は、非線型関数演算を表す（後述する図 6、図 7 参照）。

$+$ は、定数の加算を表す。ここで、 $M(4i)$ の加算は、分割したビット列の k の添字が例えば既述した図 2 の S 3 の偶数の場合に、 $i = 0, 1, 2$ のときの定数 M の値をそれぞれ加算した 3 つを出力する旨を表す。同様に、 $M(4i+1)$ 、 $M(4i+2)$ 、 $M(4i+3)$ についても、 $i = 0, 1, 2$ のときの定数 M の値をそれぞれ加算した 3 つを出力する旨を表す。

【0033】

\times は、定数の乗算を表す。ここで、 $i+1$ の乗算は、分割したビット列の k の

添字が例えば既述した図2のS4の奇数の場合に、 $i = 0, 1, 2$ のときの定数の値をそれぞれ乗算した3つを出力する旨を表す。

【0034】

+は、加算および乗算した結果の排他的論理和の演算を行なう旨を表す。

a_i から d_i は、 h 個ずつの中間データの出力を表す。ここでは、 a_i から d_i ($i = 0, 1, 2$) の3つずつの出力された中間データを表す。

【0035】

以上のように、暗号鍵を8つのグループに分割してそれぞれのビット列に非線型関数Mを演算した後、偶数のものに3つの定数をそれぞれ加算および奇数のものに3つの定数を乗算し、加算および乗算した該当データを排他的論理和演算で1つにまとめ、更に非線型関数Mを演算して3つずつの中間データを生成することが可能となる。

【0036】

図4は、本発明の説明図（拡張鍵）を示す。

図4の(a)は、 i 個ずつの中間データから段数 r をもとに1つを選択して拡大鍵を生成するシステム構成を示す。

【0037】

図4の(a)において、中間データは、ここでは、 a_i, b_i, c_i, d_i ($i = 0, 1, 2$) からなる、図3の構成で生成された中間データ（中間鍵）である。

【0038】

select 値決定装置は、作成しようとする拡大鍵の段数 r をもとに、中間データ a_i, b_i, c_i, d_i ($i = 0, 1, 2$) のうちのいずれの i のものを選択するかを決定するものである。決定は、後述する図5の(a)の式(1)に従い決定する。

【0039】

selector は、select 値決定装置によって決定された X_r, Y_r, Z_r, W_r に従い、ここでは、 $i = 0, 1, 2$ の該当するものの中間データ $a(X_r), b(Y_r), c(Z_r), d(W_r)$ を1つそれぞれ選択するもので

ある。

【0040】

データ並び替え処理装置は、段数 r をもとに、選択された中間データ $a (X_r)$ 、 $b (Y_r)$ 、 $c (Z_r)$ 、 $d (W_r)$ の並び替え（転置）を行なうものである。この転置は、後述する図5の（d）のように、段数 r に対応した転置を行なうものである。

【0041】

$G (X, Y, Z, W, r)$ 計算装置は、並び替え後の中間データ (X, Y, Z, W) をもとに、拡大鍵 $E_x Key_r$ を生成するものである。これは、後述する図4の（b）の構成で拡大鍵 $E_x Key_r$ を生成する。

【0042】

以上の構成によって、中間データから段数 r を指定して、当該段数 r の拡大鍵を生成することが可能となる。以下順次詳細に説明する。

図4の（b）は、図4の（a）の $G (X, Y, Z, W)$ 計算装置の詳細構成を示す。

【0043】

図4の（b）において、1 b i t 左巡回シフトは、データのビット列を1ビット左方向に巡回してシフトするものである。

排他的論理和は、2つのデータの排他的論理和の演算を行なうものである。

【0044】

加算は、2つのデータを加算するものである。

減算は、1つのデータから他のデータを減算するものである。

以上の回路を図示のように接続して指定段数をもとに選択および転置した後の中間データ (X, Y, Z, W) から拡大鍵を生成することが可能となる。

【0045】

図5は、本発明の説明図を示す。

図5の（a）は、 i 個ずつの中間データから段数 r のものを1つ選択するとき使用する式（1）を示す。図示の式（1）は下記である。

【0046】

$$X_r = Z_r = r \bmod 3$$

$$Y_r = W_r = r + [r/3] \bmod 3$$

図5の(b)は、図5の(a)の式(1)を模式的に示す。これは、図5の(a)の式(1)の値を実際に計算したものであって、段数 r のときに0, 1, 2の3個中から1つを選択する値であって、9個で巡回するものである。

【0047】

以上の図5の(a), (b)で段数 r に対応する値($i = 0, 1, 2$ の3個のうちの1つ)を決定し、既述した図4の(a)で i 個ずつの中間データ中から当該段数 r で決った(X_r, Y_r, Z_r, W_r)を選択することが可能となる。

【0048】

図5の(c)は、オーダ表を示す。このオーダ表は、図5の(a), (b)で選択した段数 r の中間データ(X_r, Y_r, Z_r, W_r)を並び換える(置換)するときのオーダ(順序)を決定するものである。ここでは、左側の段数 r に対応づけて右側の並び替え順序に示すように、並び替える(図4の(a)のデータ並び替え処理装置が実行する)。

【0049】

次に、図6および図7を用いて非線型関数演算について説明する。

図6の(a)は、非線型関数 M の演算の全体の構成の例を示す。ここでは、ユーザ鍵(暗号鍵) m (例えば32ビット)を入力とし、非線型関数 M を演算して結果 w (32ビット)を生成するときの様子を説明する。

【0050】

(1) ユーザ鍵の32ビットをここでは図示のように、6, 5, 5, 5, 5, 6ビットで順次 $m_0, m_1, m_2, m_3, m_4, m_5$ にそれぞれ分割する。

(2) 5ビットに分割した m_1, m_2, m_3, m_4 は、図6の(b)の $S_5(x)$ の表に従い、該当する x に対応する $S_5(x)$ の値にそれぞれ変換する。

【0051】

(3) 同様に、6ビットに分割した m_0, m_6 は、図6の(c)の $S_6(x)$ の表に従い、該当する x に対応する $S_6(x)$ の値にそれぞれ変換する。

(4) (2)と(3)によって v が図6の(a)のデータ v が生成されたこ

ととなる。

【0052】

(5) 図7の(e)に模式的に示す行列式に、図7の(d)から矢印で示す位置に置き、次に、(4)で生成したデータ v を置き、両者の行列演算を行ない、右側の w を計算する。これにより、図6の(a)のMDSを用いたXOR計算装置による結果(非線型関数 M の演算結果)が得られたこととなる。

【0053】

以上によって、図6の(a)の構成に示す順番に従い、例えば図3の非線型関数 M の演算を行なうことが可能となる。

図6の(b)は図6の(a)の $S5(x)$ のテーブル例を示し、図6の(c)は図6の(a)の $S6(x)$ のテーブル例を示す。

【0054】

図7の(c)は非線型関数 M を行列演算するときの定数(MDSを用いてXOR計算装置で用いる定数)を示し、図7の(d)はMDSを用いたXOR計算装置で行なう行列演算を模式的に示す。

【0055】

次に、既述したユーザ鍵から中間データを生成する第1段階の処理、および中間データから指定された段数 r の拡大鍵を生成する第2段階の処理について数式、記号を用いて説明する。

【0056】

(1) 第1段階の処理(ユーザ鍵から中間データを生成する処理)：

(1-1) 256ビットの暗号鍵を、32ビットごとに8個のデータ k_0, k_1, \dots, k_7 に分割する(図3参照)

(1-2) (1-1)で分割した32ビットを入力とし、32ビットを出力する非線型関数 M を用いて、以下の(1-3)から(1-6)の計算により、中間データ $a(i), b(i), c(i), d(i)$ の生成を $i=0, 1, 2$ について行なう(図3参照)。また、非線型関数 M について、(3-1)から(3-6)を実行する。

【0057】

(1-3) $a(i) = M(Ta(k0, i) \text{ XOR } Ua(k1, i))$ を計算する。ただし、 $Ta(k0, i) = M(k0) + M(4i)$, $Ua(k1, i) = M(k1) \times (i+1)$ である。尚、XORは、排他的論理和演算を表す。

【0058】

(1-4) $b(i) = M(Tb(k2, i) \text{ XOR } Ub(k3, i))$ を計算する。ただし、 $Tb(k3, i) = M(k2) + M(4i+1)$, $Ub(k3, i) = M(k3) \times (i+1)$ である。

【0059】

(1-5) $c(i) = M(Tc(k4, i) \text{ XOR } Uc(k5, i))$ を計算する。ただし、 $Tc(k4, i) = M(k4) + M(4i+2)$, $Uc(k5, i) = M(k5) \times (i+1)$ である。

【0060】

(1-6) $d(i) = M(Td(k6, i) \text{ XOR } Ud(k7, 1))$ を計算する。ただし、 $Td(k6, i) = M(k6) + M(4i+3)$, $Ud(k7, i) = M(k7) \times (i+1)$ である。

【0061】

(2) 第2段階の処理（中間データから指定された段数 r の拡大鍵を生成する処理）：

(2-1) r 段数の拡大鍵 $ExKey_r$ ($r=0, 1, \dots$) について、以下の
(2-2) から (2-4) に従い計算を行なう（図4の(a)参照）。

【0062】

(2-2) $X_r = Z_r = r \bmod 3$, $Y_r = W_r = r + [r/3] \bmod 3$
(式(1))で表される数列 X, Y, Z, W を用いて、 $(X, Y, Z, W) = (a(X_r), b(Y_r), c(Z_r), d(W_r))$ とする。

【0063】

(2-3) $r' = (r + [r/36]) \bmod 12$ を満たす r' に関して、 $(X, Y, Z, W) = \text{ORDER_12}(X, Y, Z, W, r')$ で示されるデータ並び替えを行なう。ただし、 $\text{ORDER_12}(X, Y, Z, W, r')$ は、図5の(c)に従う。

【0064】

(2-4) $ExKeyr = G(X, Y, Z, W)$ により、 r 段数の拡大鍵を計算する。ただし、 $G(X, Y, Z, W) = ((x \lll 1) + Y) \cdot XOR((Z \lll 1) - W) \lll 1$ であり、 $\lll 1$ は 1 ビット左巡回シフトを表す (図 4 の (b) 参照)。

【0065】

(3) 非線型関数 M の演算処理：

(3-1) 32 ビット入力 m から、以下の (3-2) から (3-6) に従い、32 ビットの w を出力する (図 6 の (a) 参照)。

【0066】

(3-2) m をビット分割した値 m_0, \dots, m_5 を、以下に従って与える。

$m_0 = (m \text{ の第 } 0 \text{ ビット目から第 } 5 \text{ ビット})$

$m_1 = (m \text{ の第 } 6 \text{ ビット目から第 } 10 \text{ ビット})$

$m_2 = (m \text{ の第 } 11 \text{ ビット目から第 } 15 \text{ ビット})$

$m_3 = (m \text{ の第 } 16 \text{ ビット目から第 } 20 \text{ ビット})$

$m_4 = (m \text{ の第 } 21 \text{ ビット目から第 } 25 \text{ ビット})$

$m_5 = (m \text{ の第 } 26 \text{ ビット目から第 } 31 \text{ ビット})$

(3-3) 5 ビットの入力に対し 5 ビットを出力する非線型変形関数 S_5 、6 ビットの入力に対し 6 ビットを出力する非線型変換関数 S_6 を用いて、

$s_0 = S_6(m_0)$

$s_1 = S_5(m_1)$

$s_2 = S_5(m_2)$

$s_3 = S_5(m_3)$

$s_4 = S_5(m_4)$

$s_5 = S_6(m_5)$

ここで、 S_5 、 S_6 を既述した図 6 の (b)、(c) にそれぞれ示す。

【0067】

(3-4) $v = s_0 | s_1 | s_2 | s_3 | s_4 | s_5$ を計算する。| は、ビ

ット値の連結を表す。

(3-5) v の i 番目のビット値 v_i , および 5 ビット入力から 32 ビットを出力する変換テーブル MDS を用いて、

$w = (v_0 \times \text{MDS}(0)) \text{ XOR } (v_1 \times \text{MDS}(1)) \text{ XOR } \dots \text{ XOR } (v_{31} \times \text{MDS}(31))$ を計算する。ただし、 $v_i \times \text{MDS}(i)$ は、 $v_i = 0$ のとき 0、 $v_i = 1$ のとき $\text{MDS}(i)$ である。また、MDS は図 7 の (d) に従う。

【0068】

(3-6) w を出力する。

【0069】

【発明の効果】

以上説明したように、本発明によれば、第 1 段階で暗号鍵から中間データを生成し、第 2 段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成する構成を採用しているため、拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることが可能となる。これにより、

(1) 例えば中間データを 1 個生成するためには大きな処理時間が必要であるがあるが、拡大鍵生成装置 5 により、必要な中間データの個数を少なくでき、安全性の高い拡大鍵を高速に生成できる。

【0070】

(2) また、生成した拡大鍵 $ExKey_0, ExKey_1, \dots, ExKey_{n-1}$ の全てを記憶せずに、暗号化あるいは復号化の処理の途中で、必要となる拡大鍵のみを生成する場合、指定した段数 r の拡大鍵のみを高速生成できるという顕著な特徴がある。この効果を以下説明する。

【0071】

一般的に共通鍵暗号方式は、暗号化で $ExKey_0, ExKey_1, \dots, ExKey_{n-1}$ の順に拡大鍵が用いられる場合、復号化では、 $ExKey_{n-1} \dots ExKey_1, ExKey_0$ のように、暗号化とは逆の順で拡大鍵が用いられる。ここで、 $ExKey_1$ を生成するのに $ExKey_0$ の値を必要とする拡大鍵生成方式を用いて (既述した図 9 参照)、逐次生成を行なった場合、 Ex

Key 1 は直接生成することができず、先に ExKey 0 を生成してこれを用いて ExKey 1 を生成することとなり、その分、復号化の拡大鍵生成時間が暗号化よりも遅くなる。

【 0 0 7 2 】

これに対し、本発明は、他の拡大鍵とは独立して、任意の段数 r を指定して拡大鍵を生成できるため、拡大鍵を ExKey 0, ExKey 1, \dots ExKey $n-1$ の順に生成するのも、ExKey $n-1 \dots$ ExKey 1, ExKey 0 の順に生成するのも、同じ処理時間で行なうことができるという顕著な特徴がある。

【 0 0 7 3 】

以上のように、本発明では、拡大鍵の逐次生成を行なう場合でも、暗号化と復号化の処理時間を同じにして、復号化の拡大鍵生成時間を暗号化よりも遅くなることを防ぐ顕著な効果がある。

【図面の簡単な説明】

【図 1】

本発明のシステム構成図である。

【図 2】

本発明の動作説明フローチャートである。

【図 3】

本発明の説明図（中間データ）である。

【図 4】

本発明の説明図（拡大鍵）である。

【図 5】

本発明の説明図である。

【図 6】

本発明の説明図（非線型関数演算、その 1）である。

【図 7】

本発明の説明図（非線型関数演算、その 2）である。

【図 8】

共通鍵方式の説明図である。

【図 9】

従来の DES のアルゴリズムの全体のブロック図である。

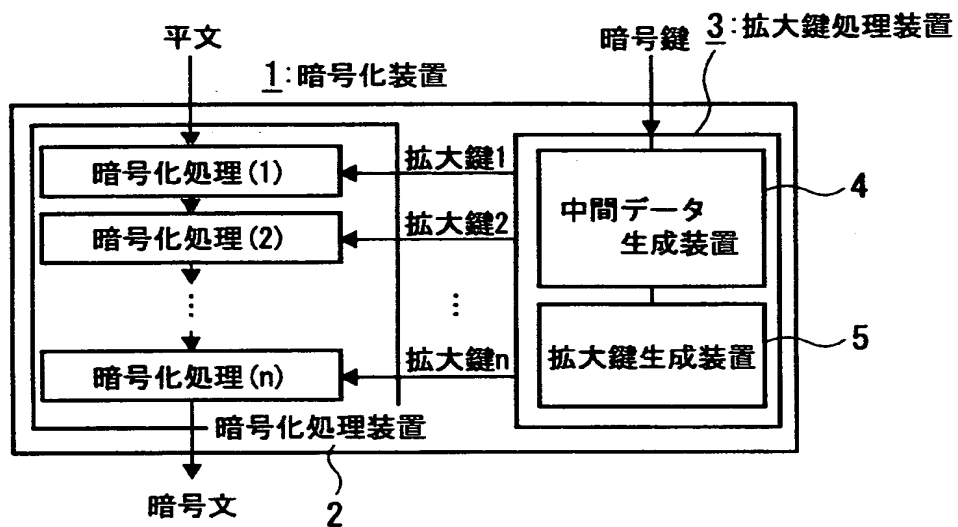
【符号の説明】

- 1 : 暗号化装置
- 2 : 暗号化処理装置
- 3 : 拡大鍵処理装置
- 4 : 中間データ生成装置
- 5 ; 拡大鍵生成装置
- M : 非線型関数
- G : 非可逆関数

【書類名】 図面

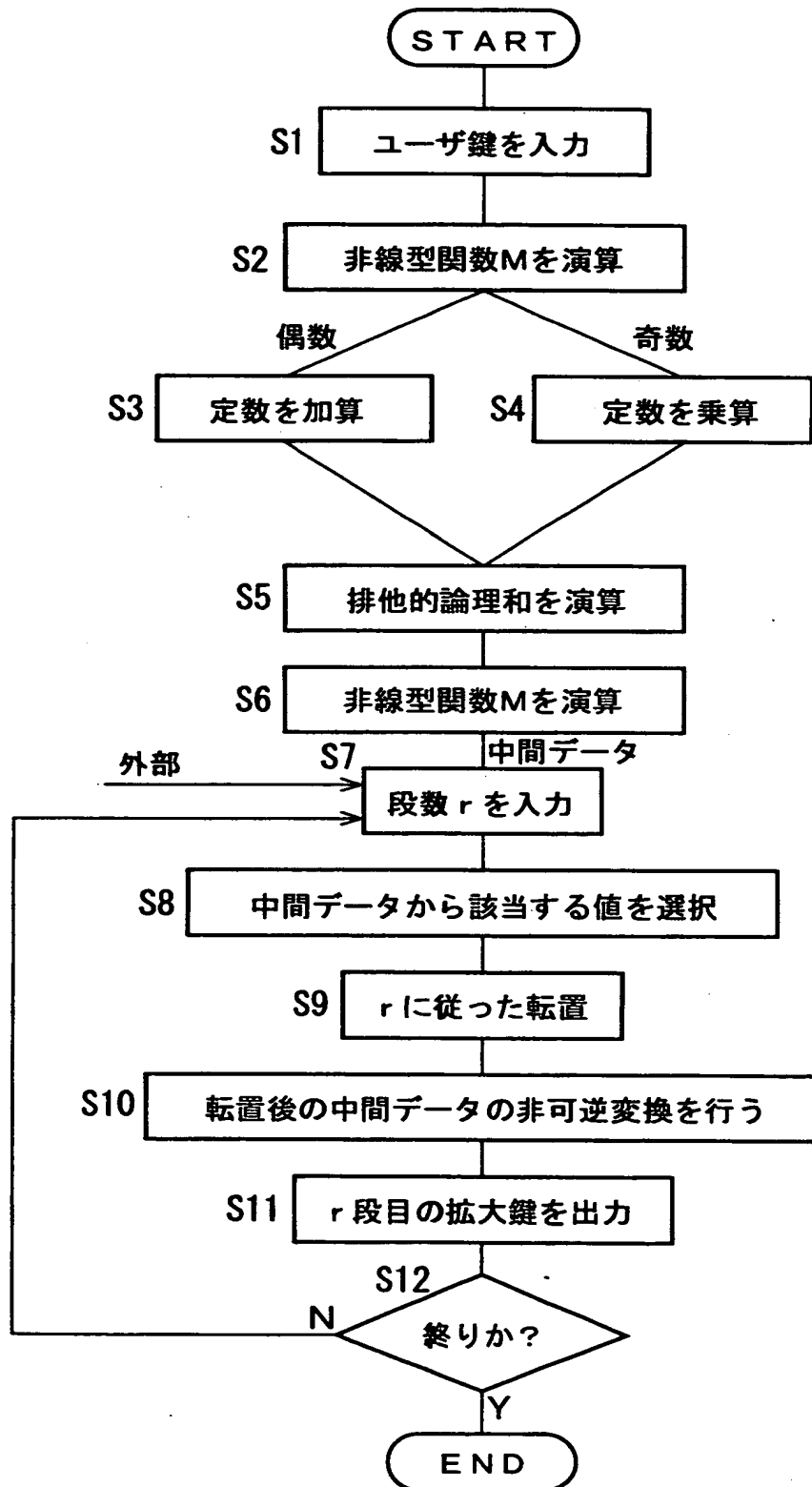
【図 1】

本発明のシステム構成図

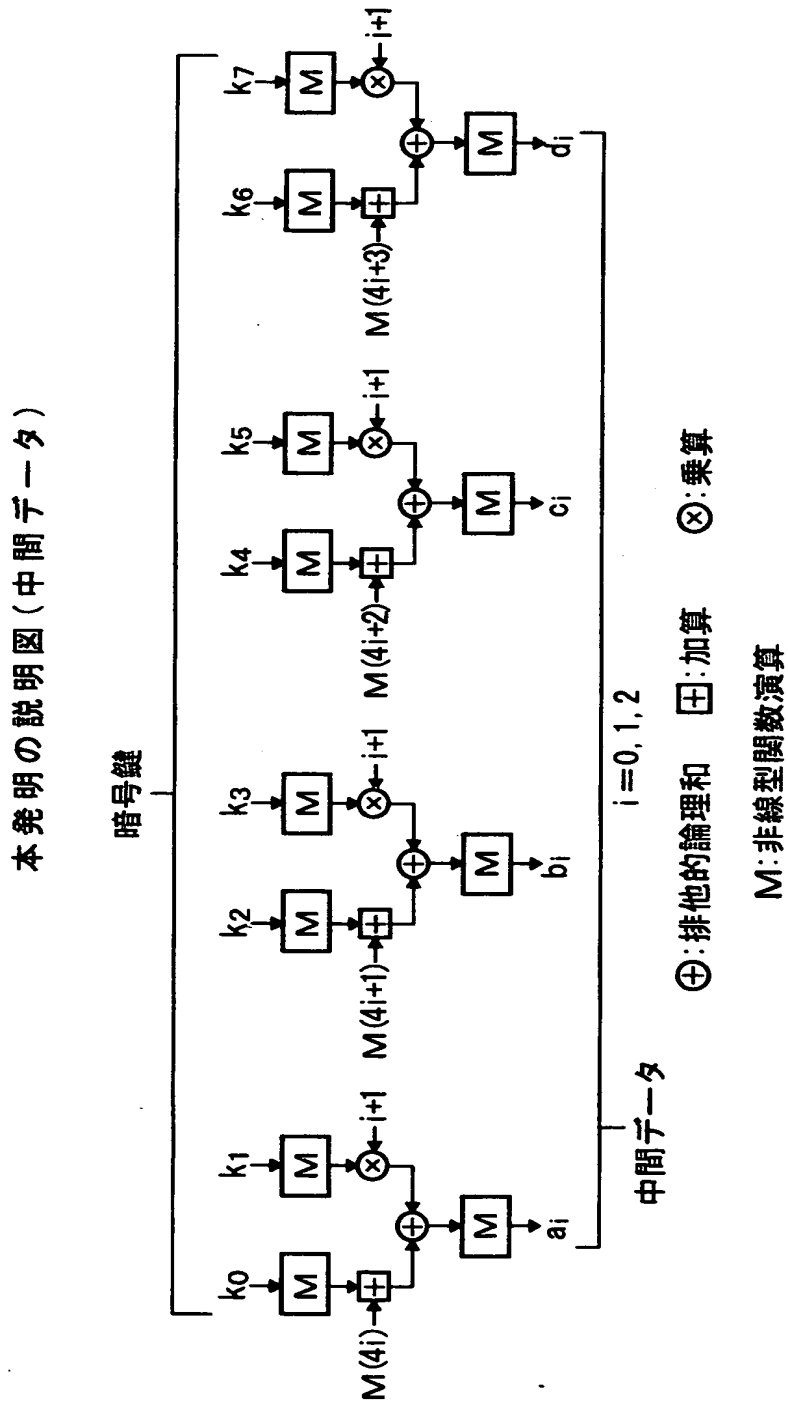


【図2】

本発明の動作説明フローチャート

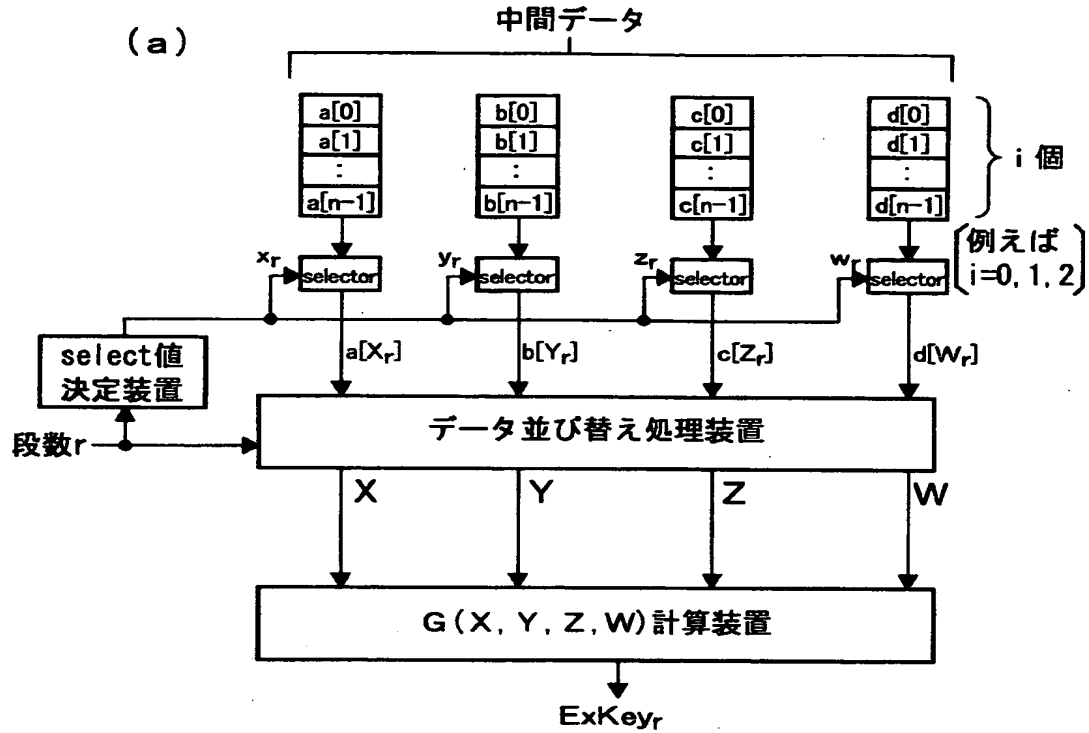


【図3】

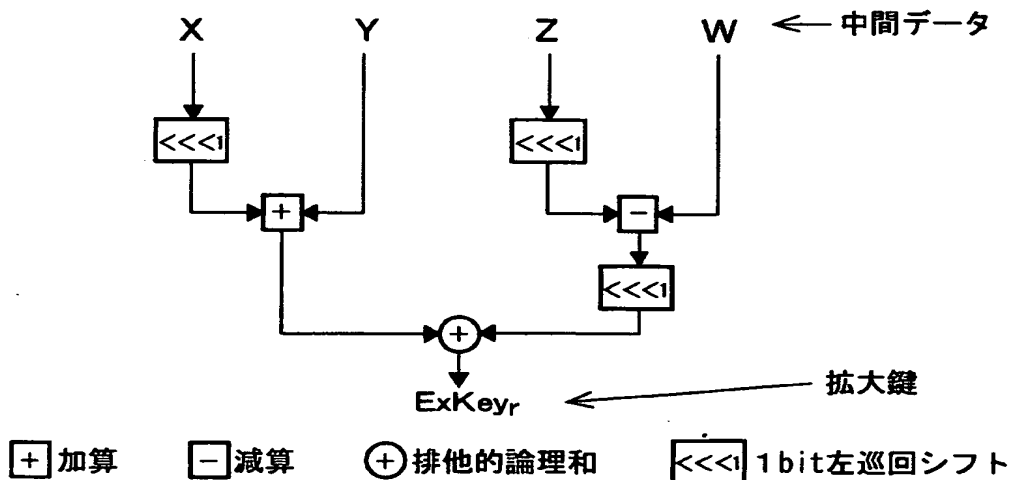


【図 4】

本発明の説明図(拡大鍵)



(b) $G(X, Y, Z, W)$ 計算装置



【図5】

本発明の説明図

(a) $x_r = z_r = r \bmod 3, y_r = w_r = r + [r/3] \bmod 3 \dots$ 式(1)

(b) 9個で巡回

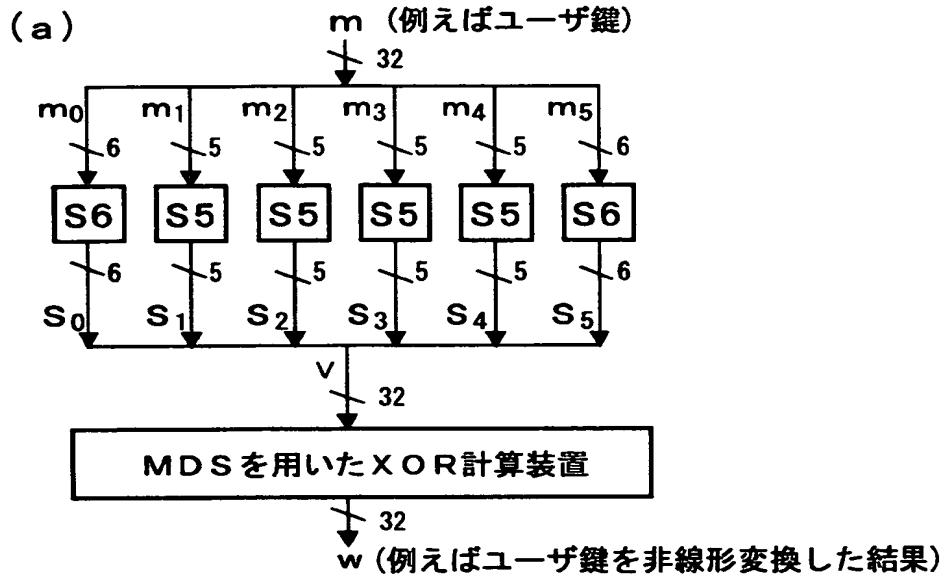
r	0	1	2	3	4	5	6	7	8	9	0...
X _r	0	1	2	0	1	2	0	1	2	0	1...
Y _r	0	1	2	1	2	0	2	0	1	0	1...
Z _r	0	1	2	0	1	2	0	1	2	0	1...
W _r	0	1	2	1	2	0	2	0	1	0	1...

(c) オーダ表

r (段目)	ORDER 12(X,Y,Z,W,r)	並び替え
0	(X,Y,Z,W)	
1	(Y,X,W,Z)	
2	(Z,W,X,Y)	
3	(W,Z,Y,X)	
4	(X,Z,W,Y)	
5	(Y,W,Z,X)	
6	(Z,X,Y,W)	
7	(W,Y,X,Z)	
8	(X,W,Y,Z)	
9	(Y,Z,X,W)	
10	(Z,Y,W,X)	
11	(W,X,Z,Y)	

【図 6】

本発明の説明図（非線型関数演算，その 1）



(b) S5(x)

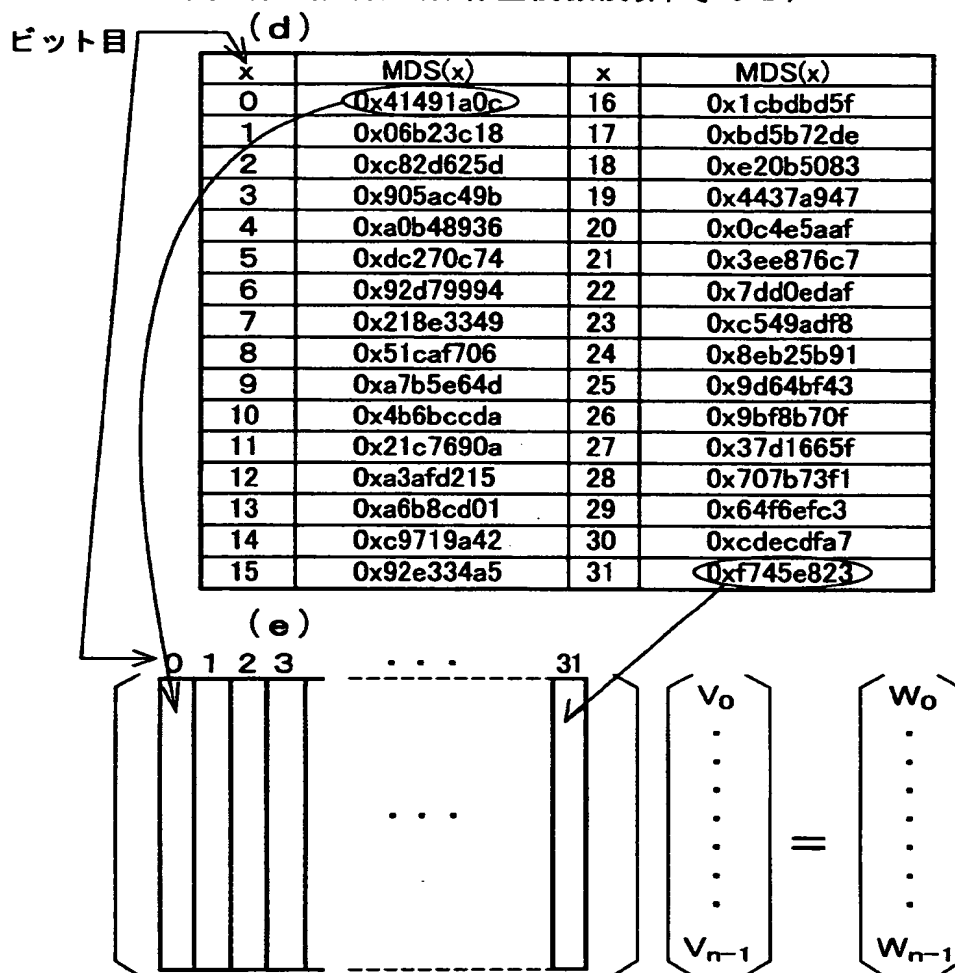
x	S5(x)	x	S5(x)	x	S5(x)	x	S5(x)
0	20	8	22	16	27	24	23
1	26	9	30	17	11	25	5
2	7	10	13	18	1	26	8
3	31	11	14	19	21	27	3
4	19	12	4	20	6	28	0
5	12	13	24	21	16	29	17
6	10	14	9	22	2	30	29
7	15	15	18	23	28	31	25

(c) S6(x)

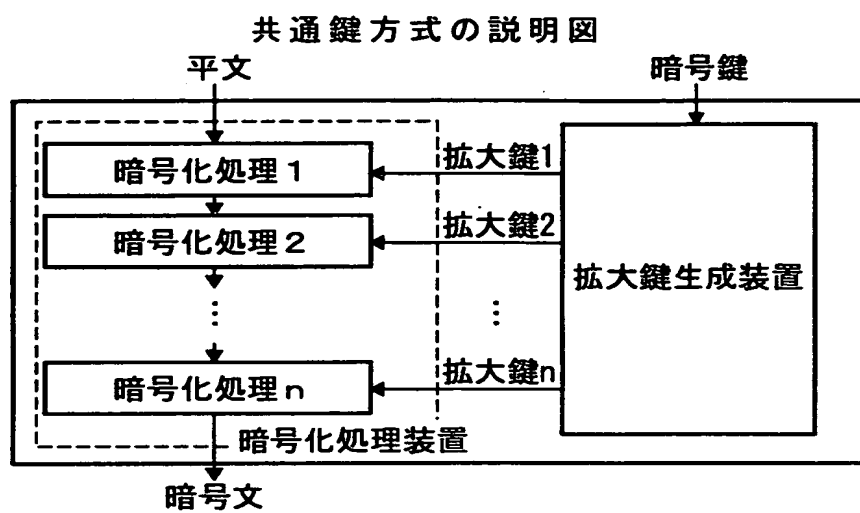
x	S6(x)	x	S6(x)	x	S6(x)	x	S6(x)
0	47	16	37	32	62	48	3
1	59	17	63	33	52	49	16
2	25	18	20	34	35	50	41
3	42	19	61	35	18	51	34
4	15	20	55	36	14	52	33
5	23	21	2	37	46	53	7
6	28	22	30	38	0	54	45
7	39	23	44	39	54	55	49
8	26	24	9	40	17	56	50
9	38	25	10	41	40	57	58
10	36	26	6	42	27	58	1
11	19	27	22	43	4	59	21
12	60	28	53	44	31	60	43
13	24	29	48	45	8	61	57
14	29	30	51	46	5	62	32
15	56	31	11	47	12	63	18

【図 7】

本発明の説明図(非線型関数演算, その 2)

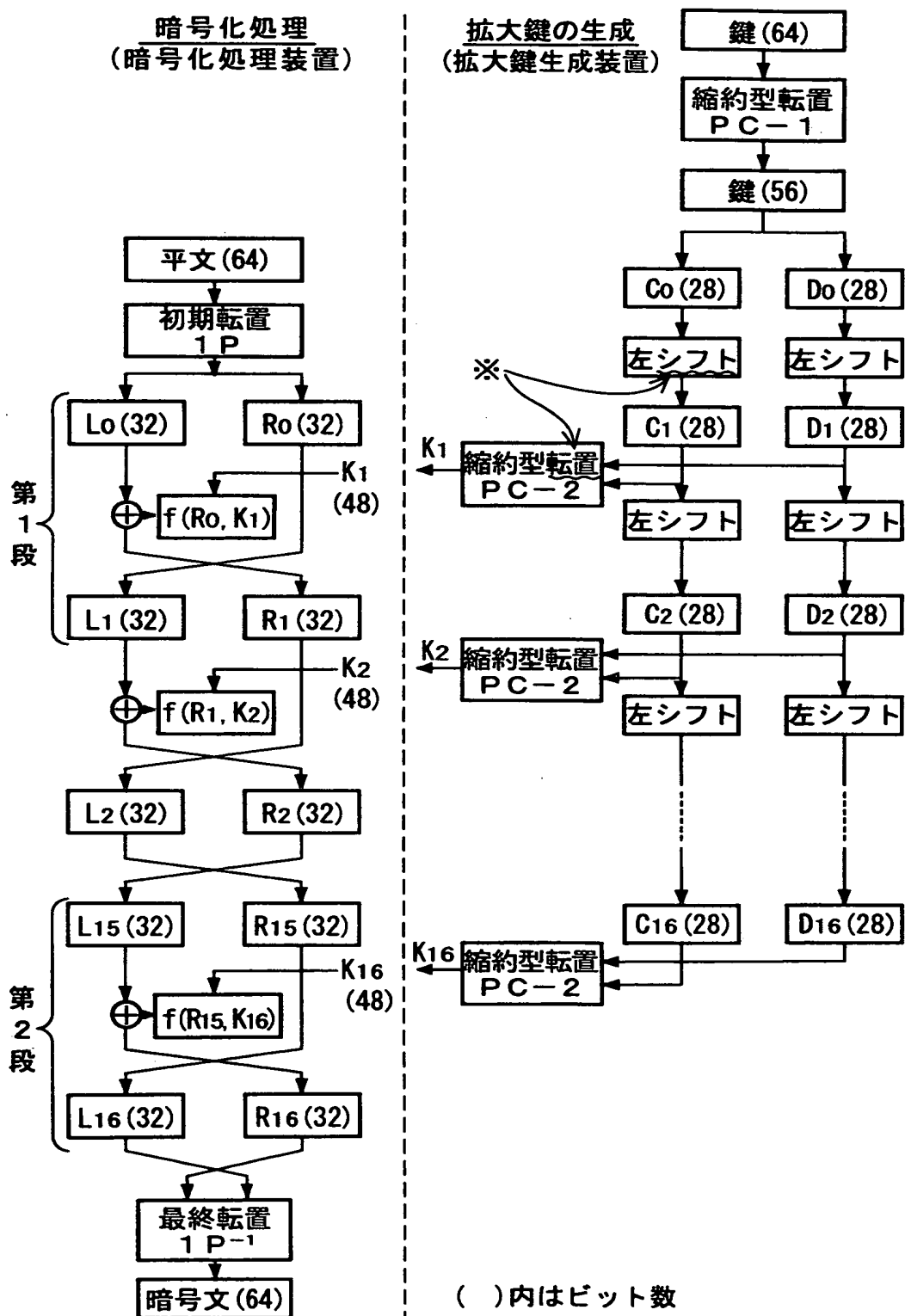


【図 8】



【図 9】

従来の DES のアルゴリズムの全体のブロック図



【書類名】 要約書

【要約】

【課題】 本発明は、暗号鍵から拡大鍵を生成する拡大鍵生成装置および記録媒体に関し、第1段階で暗号鍵から中間データを生成し、第2段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成し、任意段の拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることを目的とする。

【解決手段】 入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数 i 回それぞれ行なって複数 i の演算結果を生成し、これら生成した各グループ毎の複数 i の演算結果について複数のグループ間で該当演算結果をそれぞれ1つにまとめる演算を行ない、複数 i の中間データを生成する中間データ生成手段と、指定された拡大鍵の段数 r をもとに、複数 i の中間データから1つを選択し、選択した中間データを非可逆変換して段数 r の拡大鍵を生成する拡大鍵生成手段とを備えるように構成する。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社